

## HOMEWORK #4:

### *All of Mom's Robots*

Due October the 17th, 11:59pm

For this assignment, your `main()` function should be in a C++ file called `allrobots.cpp`.

Remember to put your name and section at the top of your program file.

Your program should expect all input to come from `'cin'`, and all your output should be to `'cout'`.

#### Problem

MomCorp, the largest robot manufacturing company in the planet, has noticed your great programming skills and now wants to hire you for a special project. During the annual “Mother’s day” event, all robots gather together to celebrate their creator. MomCorp will conduct a census and gather statistics on all the robots that gather there. Therefore, they have hired you to write a program to help them analyse the census data.

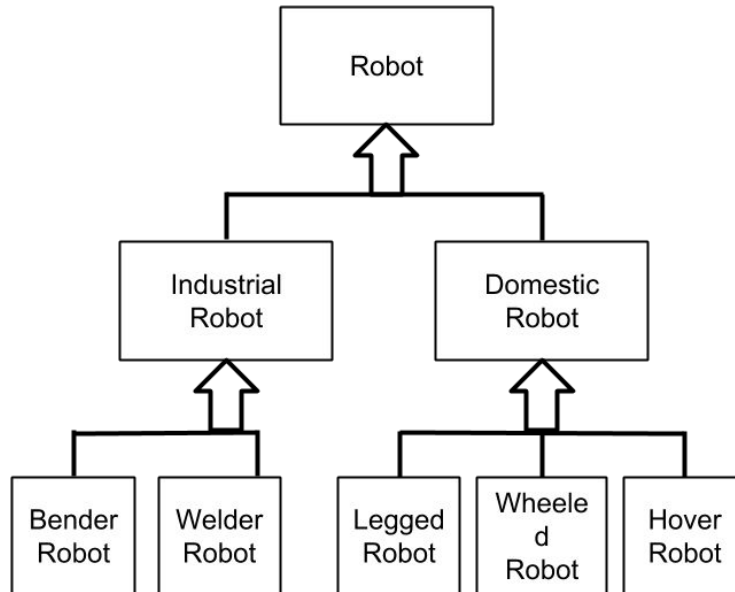


“I only want the best for my beloved Robots”

## Kinds of Robots:

The MomCorp Robot Manufacturing company makes both **Industrial** and **Domestic** Robots. **Industrial** robots come in 2 kinds: **Bender** and **Welder** robots. Domestic robots come in 3 kinds: **Legged**, **Wheeled** and **Hover** robots.

If you were to order the different kinds onto a *class hierarchy*, it will look like:



The class hierarchy of robots.

## Robot Attributes:

- **Robot:** Every robot has a name.
  - **Industrial Robots** are equipped with an integer number of power sources
    - **Bender Robots** may be equipped with an integer number of geometric coprocessors.
    - **Welder Robots** have an integer number of armor plates, and an armor density (given as a floating number).
  - **Domestic Robots** have one or more cpu's and one or more "Application Behaviors" (called simply "apps")
    - **Legged Robots** have an even number of legs.
    - **Wheeled Robots** have one or more wheels.
    - **Hover Robots** come equipped with an integer number of rockets and an integer number of anti-gravity engines.

## Robot Metrics:

Every robot can be evaluated based on 4 metrics:

1. *Speed*: measured in circus-rings per second.
2. *Strength*: given in dinosaur-power (There is no horses in the 31st century)
3. *Toughness*: measured in the number of hits from a Miley Cyrus' wrecking ball a robot can stand before malfunction.
4. *Intelligence*: measured in nano-Turings.

Each different kind of robot can be evaluated for each metric as follows:

- **Bender Robots:**

**Speed**: equal to  $(0.3 * \text{number of power sources}) + (0.02 * \text{number of geometric coprocessors})$

**Strength**: equal to  $0.55 * \text{number of power sources}$

**Toughness**: equal to 3.1415

**Intelligence**: equal to  $0.34 + (1.3 * \text{number of geometric coprocessors})$

- **Welder Robots:**

**Speed**: equal to  $(0.1 * \text{number of power sources}) - (0.22 * \text{number of armor plates} * \text{armor density})$

**Strength**: equal to  $(3.15 * \text{number of power sources})$

**Toughness**: equal to  $(5.19 * \text{number of armor plates} * \text{armor density})$

**Intelligence**: equal to 0.45

- **Legged Robots:**

**Speed**: equal to  $7.15 - (0.04 * \text{number of legs}^2)$

**Strength**: equal to 0.123

**Toughness**: equal to  $1.4 * (1 / \text{number of legs})$

**Intelligence**: equal to  $2.4 * (\text{number of CPU's} / \text{number of apps})$

- **Wheel Robots:**

**Speed**:  $0.33 * (\text{number of wheels}^2)$

**Strength**: 4.444

**Toughness**:  $5.12 - (0.7 * \text{number of wheels})$

**Intelligence**:  $1.3 * (\text{number of CPU's} / \text{number of apps})$

- **Hover Robots:**

**Speed**:  $0.59 * \text{number of rockets} * \text{number of anti-gravity engines}$

**Strength:** equal to  $1.12 * (1 / \text{number of anti-gravity engines})$   
**Toughness:** equal to 0.99  
**Intelligence:**  $5.2 * (\text{number of CPU's} / \text{number of apps})$   
 $- 0.8 * (1 / \text{number of rockets})$

## Input

The first line of the input gives the number of robots in the census file **R**.

**R** lines will follow, each describing a robot's vital statistics.

The first line of each robot is an integer stating the robot's type, followed by a string stating the robot's name. Then, for Industrial Robots the number of power sources, or for domestic robots the number of CPU's and the number of "Apps".

Finally, according to the type of the robot:

For Bender Robots, the number of geometric coprocessors.

For Welder Robots, the number of armor plates and the armor plate density.

For Legged Robots, the number of legs.

For Wheeled Robots, the number of wheels.

For Hover Robots, the number of rockets and the number of anti-gravity engines.

The robot type numbers are:

0 - Bender 1 - Welder 2 - Legged 3 - Wheeled 4 - Hover
--

## Output

Output a report on the overall statistics of the census data as shown in the sample output.

For each metric, output the average value over all the robots, and report the name of the robots with the highest and lowest values.

## Implementation Requirements

- You are provided with an abstract base class 'Robot'.  
Create derived classes for each different kind of robot as shown in the class hierarchy.  
(7 classes total)  
Consider what functionality is needed for each kind.
- Do not use any public member variables. ( respect the privacy of the robots ).
- Use **static const** for the necessary constants of each class.

- The names of the robots will **not** contain spaces.

## Sample

Input				
10				
0	Flexo	2	2	
1	Sparky	5	4	0.45
2	Shaky	1	1	2
3	Turtle	1	3	3
4	Vincent	5	10	4 2
0	Twisty	1	1	
1	Weldomatic	8	8	1.68
2	Spidey	2	4	8
3	Jason	2	1	6
4	Bob	4	4	1 1

Output				
[Speed]				
The average speed is 3.28 circus-rings per second.				
The highest speed value is 11.88 by Jason.				
The lowest speed value is 0 by Weldomatic.				
[Strength]				
The average strength is 5.341 dinosaur-power.				
The highest strength value is 25.2 by Weldomatic.				
The lowest strength value is 0.123 by Shaky.				
[Toughness]				
The average toughness is 9.217 wrecking-balls.				
The highest toughness value is 69.75 by Weldomatic.				
The lowest toughness value is 0.175 by Spidey.				
[Intelligence]				
The average intelligence is 1.891 nano-Turings.				
The highest intelligence value is 4.4 by Bob.				
The lowest intelligence value is 0.4333 by Turtle.				

**Addendum:**

- Format float number output using `#include<iomanip>` and `cout << setprecision(4).`
- The speed of a robot cannot be negative. Robots whose speed value is negative have speed 0.0. (They are either too heavy or have too many legs to move)
- In case of a tie between two robots for any of the metrics, output the robot that appears the earliest in the list.